"Express Mail" Mailing Label No. **EL436467436US**

PATENT APPLICATION
ATTORNEY DOCKET NO. OR99-17501

FACILITATING SINGLE SIGN-ON BY USING AUTHENTICATED CODE TO ACCESS A PASSWORD STORE

Inventor(s): Vipin Samar

Related Application

Sub A1

~~The subject matter of this application is related to the subject matter in a~~ co-pending non-provisional application by the same inventor as ~~the~~ instant application and filed on the same day as the instant ~~application~~ entitled, "Method and Apparatus for Facilitating ~~Single~~ Sign On through Redirection to a Login Server," having serial number TO BE ASSIGNED, and filing date TO BE ~~ASSIGNED (Attorney Docket No. OR99-17601).~~

BACKGROUND

Field of the Invention

The present invention relates to security in a distributed computing environment. More specifically, the present invention relates to a method and an

apparatus for facilitating a single sign on to multiple applications by using applets (or other code fragments, modules or plug-ins) to access a password store.

Related Art

With the recent proliferation of web-based applications, the number of remote applications a typical user accesses has grown dramatically. For security purposes, it is often necessary to authenticate a user before allowing the user to access certain applications. This type of authentication is most commonly accomplished by requiring the user to provide a password for each application. This solution has been generally satisfactory until recently because users have typically accessed only a small number of applications.

However, this solution becomes less satisfactory when a large number of applications are involved. This is because it is extremely burdensome for a user to have to enter dozens of passwords each day. Furthermore, the proliferation in applications requiring passwords tends to compromise security because a user is typically unable to remember dozens of different passwords for dozens of different applications.

In order to keep track of different passwords, a user can write down all of the different passwords on yellow sticky notes attached to a computer monitor. However, writing passwords down in this way can greatly compromise security.

More typically, a user uses a single password for all of the different applications the user accesses. This creates even more of a security problem because this single password is known by numerous applications running on numerous computing systems. If any one of these applications or computer systems is insecure, the secrecy of the single password can be compromised.

Furthermore, as the number of passwords proliferate, help desks become burdened with requests to deal with forgotten or misplaced passwords, which can increase the cost of administering applications.

Additionally, users tend to use the simplest and shortest password possible in order to reduce the time required to enter the password and to make the password easy to remember. However, these shorter and simpler passwords tend to be less random and can be more easily cracked.

One solution to the authentication problem is to employ the public key infrastructure (PKI) to authenticate a user to various applications. PKI makes use of public key-private key pairs and chains of digital certificates to authenticate a user to an application. However, PKI has yet to be widely adopted because solutions to technical problems relating to certificate management and key life-cycle management are still being developed. Furthermore, it is difficult to retrofit legacy applications to make use of PKI.

Another solution to the authentication problem is to provide a single sign on facility. In a conventional single sign on facility, a user's passwords are stored in a single password store protected by login authentication or by operating system authentication. When an application is run, it retrieves a password associated with the application from the password store.

However, the problem with using a conventional password store is that it is possible for a rogue application to read the entire password store. Hence, users must completely trust all of the applications that have access to the password store.

What is needed is a method and an apparatus for providing a single sign on facility that does not require the applications that make use of the single sign on facility to be completely trusted.

SUMMARY

One embodiment of the present invention provides a system that facilitates accessing to a plurality of applications that require passwords. When the system receives a request for a password from an application running on a remote computer system, the system first authenticates the request to ensure that it originated from a trusted source. Next, the system uses an identifier for the application to look up the password for the application in a password store, which contains passwords associated with the plurality of applications. If the password exists in the password store, the system sends the password or a function of the password to the application on the remote computer system. Hence, the system creates the illusion that there is a single sign on to a large number of applications, whereas in reality the system automatically provides different passwords to the applications as they are requested.

Sub B1 ~~In one embodiment of the present invention, the request for the password~~ includes computer code that when run on the local computer system requests the password on behalf of the application on the remote computer system. In a variation on this embodiment, the computer code is in the form of a downloadable piece of software or an installed piece of software that runs in the execution environment of the local machine. In a variation on this embodiment, the computer code is in the form of a JAVA applet that runs on a JAVA virtual machine on the local computer system. In a variation on this embodiment, sending the password or the function of the password to the application to the remote computer system involves communicating the password to the JAVA applet, and allowing the JAVA applet to forward the password to the application ~~on the remote computer system~~.

Sub B2 In one embodiment of the present invention, the JAVA applet is a signed JAVA applet, and authenticating the request involves authenticating the JAVA applet's certificate chain.

In one embodiment of the present invention, authenticating the request involves authenticating the creator of the request.

In one embodiment of the present invention, authenticating the request involves authenticating the remote computer system that sent the request.

In one embodiment of the present invention, if the password store is being accessed for the first time, the system prompts a user for a single sign on password, and uses the single sign on password to open the password store.

In one embodiment of the present invention, if a time out period for the password store expires, the system prompts the user again for the single sign on password for the password store, and then uses the single sign on password to open the password store.

In one embodiment of the present invention, if the password store is being accessed for the first time, the system authenticates the user through an authentication mechanism. This authentication mechanism can include a smart card, a biometric authentication mechanism or a public key infrastructure.

In one embodiment of the present invention, if the password does not exist in the password store, the system adds the password to the password store, and then sends the password to the application on the remote computer system. In a variation on this embodiment, adding the password to the password store involves automatically generating the password. In another variation on this embodiment, adding the password to the password store involves asking a user to provide the password.

In one embodiment of the present invention, the system additionally decrypts data in the password store prior to looking up the password in the password store.

In one embodiment of the present invention, the password store is located on a second remote computer system.

In one embodiment of the present invention, the password store is located on a local smart card, a floppy disk, or a memory button.

In one embodiment of the present invention, the system receives a request to change the password from the application on the remote computer system. In response to this request, the system automatically generates a replacement password and stores the replacement password in the password store. Next, the system forwards the replacement password or the password function to the application on the remote computer system.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 illustrates a distributed computer system in accordance with an embodiment of the present invention.

FIG. 2 illustrates the structure of an entry in a password store in accordance with an embodiment of the present invention.

FIG. 3 is a flow chart illustrating the process of facilitating a single sign on in accordance with an embodiment of the present invention.

FIG. 4 is a flow chart illustrating the process of automatically generating a replacement password in accordance with an embodiment of the present invention.

FIG. 5 is a flow chart illustrating the process of opening the password store after a time out period has expired in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

The data structures and code described in this detailed description are typically stored on a computer readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital video discs), and computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated). For example, the transmission medium may include a communications network, such as the Internet.

Distributed Computing System

FIG. 1 illustrates a distributed computer system 100 in accordance with an embodiment of the present invention. Distributed computer system 100 includes a number of clients, including client 101. Client 101 communicates across network 140 with servers 171-173 and database server 150. Client 101 can include any node on network 140 including computational capability and including a

mechanism for communicating across network 140 with servers 171-173 and database server 150. Network 140 can include any type of wire or wireless communication channel capable of coupling together computing nodes. This includes, but is not limited to, a local area network, a wide area network, or a combination of networks. In one embodiment of the present invention, network 140 includes the Internet. Servers 171-173 can include any nodes on network 140 including a mechanism for servicing requests from client 101 for computational and/or data storage resources. Database server 150 can include any node on network 140 including a mechanism for processing database access requests.

Sub A2

Client 101 includes browser 130. Browser 130 can include any type of web browser capable of viewing a web site, such the INTERNET EXPLORER™ browser distributed by the Microsoft Corporation of Redmond, Washington.

Sub B3

In the embodiment illustrated in FIG. 1, web browser 130 includes a JAVA VIRTUAL MACHINE™ (JVM) 120 that executes platform-independent instructions in the form of JAVA™ applets comprising JAVA bytecodes. (The terms JAVA, JAVA VIRTUAL MACHINE and JAVA DEVELOPMENT KIT are trademarks of SUN Microsystems, Inc. of Palo Alto, California.)

Sub B4

JVM 120 executes a number of applets 171-173 that originate from applications 161-163, respectively, located on servers 171-173, respectively. Note that applets 171-173 are "signed applets." A signed applet includes a digital signature, which is generated and verified through use of a private key-public key pair.

The JAVA DEVELOPMENT KIT™ (JDK) 1.2 (distributed by SUN Microsystems, Inc. of Palo Alto, CA) provides a facility to authenticate a signed applet to verify whether the signed applet originates from a trusted source. This facility includes mechanisms to perform authentication through a chain of digital certificates. This authentication facility is generally contained within signature

verifier 106 in FIG. 1. JDK 1.2 gives a signed applet access to system resources selectively, through a permissions model that allows fine-grained accesses to system resources.

Sub B5

~~JVM 120 additionally includes a bytecode verifier 104 that verifies that bytecodes within applets 171-173 are properly formed.~~

Sub B6

~~Applets 171-173 request passwords on behalf of applications 161-163,~~ respectively, on servers 171-173, respectively. These requests are directed to password lookup mechanism 103, which looks up the requested passwords in password store 102. In one embodiment of the present invention, password store 120 is located on client 101. In another embodiment, password store 102 is located within storage device 121 coupled to database server 150. In this embodiment, a user is able to access his or her password store from any location ~~on network 140.~~

Sub B7

~~In one embodiment of the present invention, database server 150 can be~~ accessed by using commands adhering to the lightweight directory access protocol ~~(LDAP). Password store 120 can alternatively be stored on a smart card.~~

Password Store Entry

Sub B8

~~FIG. 2 illustrates the structure of an entry 200 within password store 102~~ in accordance with an embodiment of the present invention. Entry 200 includes a number of fields, including service name 202, download point 203, user ID 204, password 205, expiry time 206, last login time 207 and last update time 108. Service name 202 includes an identifier for the service (or application) that is requesting the password. Download point 203 contains an identifier for the point from which the applet is downloaded to client 101. For example, in FIG. 1 applet 181 is downloaded from server 171. User ID 204 specifies the name of the user that is requesting access to applications 161-163. Password 205 specifies the

password associated with user ID 204. Note that user ID 204 is typically sent along with password 205 or a function of the password to an application. Expiry time 206 specifies an expiration time for password 205. Last login time 207 specifies when the user associated with user ID 204 was last logged in to the application specified by service name 202. Last update 208 specifies when password store entry 200 was last updated.

Single Sign On Process

Sub B9 10

FIG. 3 is a flow chart illustrating the process of facilitating a single sign on in accordance with an embodiment of the present invention. The system first prompts the user for a single sign on password (step 302), and then uses the single sign on password to open password store 102 (step 304). In one embodiment of the present invention, the process of opening password store 102 includes using the single sign on password to decrypt the password store. Note that the process of opening password store 102 can take place during system initialization. Alternatively, it can take place the first time password store 102 is accessed after system initialization.

Next the system receives an applet from an application on a server (step 306). For example, client 101 can receive an applet 181 from application 161 on server 171.

JVM 120 authenticates a digital signature and certificate chain for applet 181 (step 308). Note that JDK 1.2 is presently configured to perform such authentication. This authentication gives JVM 120 a very high degree of confidence that applet 181 originated from application 161. Without such authentication, client 101 may give out a password to a rogue application that pretends to be application 161.

Sub B10

~~Next, JVM 120 executes applet 161. During this execution, applet 181~~ requests a password from client 101 on behalf of application 161. This request is ~~received by client 101 (step 310).~~

Also note that this authentication can involve authenticating the machine (download point) that sent applet 181, which in this case is server 171. The authentication can also involve authenticating the application that sent applet 181, which in this case is application 161.

As an alternative to authentication through digital certificates, the system can instead verify that applet 181 originated from a specific uniform resource location (URL) or Internet protocol (IP) address. This verification gives the system some measure of confidence that applet 181 originated from a trusted source. However, this alternative is not as secure as authenticating a digital signature.

Sub B11

~~Next, the system uses user ID 204 and service name 202 to look up~~ password 205 in password store 102 (step 312). If password 205 does not exist in password store 102, the system adds password 205 to password store 102 (step 316). This may involve prompting the user for password 205, and subsequently ~~adding the password 205 to password store 102.~~

Finally, the system communicates password 205 (and possibly user ID 204) to applet 181 (step 318), and allows applet 181 to forward password 205 to application 161 on server 171 (step 320). At this point, the user can communicate with application 161 so that application 161 can perform whatever work is required by the user (step 321).

Sub B12

~~Note that the present invention is not limited to using JAVA applets to~~ perform the password lookup. Other embodiments of the present invention use other types of code including ACTIVEX™ and signed ACTIVEX code. (ActiveX ~~is a Trademark of the Microsoft Corporation of Redmond, Washington).~~

Process of Generating Replacement Password

FIG. 4 is a flow chart illustrating the process of automatically generating a replacement password in accordance with an embodiment of the present invention. This happens commonly because many applications require passwords to be periodically changed for security purposes.

Sub B13

The system starts by receiving a request to change a password from an application, such as application 161 on server 171 (step 402). In response to this request, the system automatically generates a replacement password (step 404). Since the computer system (and not a human being) generates the replacement password, the replacement password can be quite long and quite random, which makes the password more secure.

In one embodiment of the present invention, the request is received in the form of a password update applet from application 161. This applet is authenticated as is described above with reference to step 308.

The system then stores the replacement password in the associated entry within password store 102 (step 406), and then forwards the replacement password to application 161 on application server 171 (step 408).

Sub B14

As part of sending the replacement password to application 171, the system may additionally send the old password (or a function of the old password) so that application 171 can verify that the entity that generated the replacement password was in possession of the old password.

Process of Opening Password Store

Sub B15

FIG. 5 is a flow chart illustrating the process of opening password store 102 after a time out period has expired in accordance with an embodiment of the present invention. The process starts when a time out period of password store

102 expires (step 502). In response to this expiration, the system again prompts the user for the single sign on password (step 504). The system uses the single ~~sign on password to open the password store (step 506).~~

This time out period can be set by the user. For example, the user can set the time out period to be nine hours. In this case, when the user signs on in the morning, this single sign on will be good for the rest of the business day. This prevents somebody from coming into the user's office during the night and using the active single sign on session to make unauthorized accesses the all of the applications that are covered by the single sign on system.

The foregoing descriptions of embodiments of the invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.